

### **Remarks**

Reconsideration of the application is respectfully requested in view of the foregoing amendments and following remarks. Claims 1-32 are pending in the application. Claims 1, 6, 11, 18, 21, 22, 25, 28, and 29 are independent.

#### ***Patentability Over “International Layout in CSS” and “Mastering Excel 97”***

The Office has asserted a rejection of claims 1-32 as obvious over “International Layout in CSS”, 01/1999, W2C, pages 1-40 (herein after “CSS”) in view of Chester et al., “Mastering Excel 97”, 1997, Fourth Edition, Sybex, Inc., pages 129-134 (herein after “Excel”). Applicants respectfully traverse.

#### ***Claim 1***

***The proposed combination fails to teach or suggest anything about “a minimum width input designed to be used by a table cell element with horizontal character flow [but] the given table cell has vertical character flow.”***

Applicants respectfully assert that the Office has failed to carry the burden of establishing a prima facie case of obviousness, because the art of record fails to teach or suggest “*determining that a call to a table cell sizing method from a table sizing method is a request to determine an acceptable minimum width table cell measurement for a given table cell and the request includes a minimum width input designed to be used by a table cell element with the horizontal character flow property to render a minimum table cell width; [and] determining that the given table cell content element has the vertical character flow property*”.

Specifically, claim 1 recites,

1. (original) In a computer system with a textual mark-up language engine, a table sizing method designed to auto-size table elements, a table cell sizing method designed to auto-size table cell elements with a horizontal character flow property, a method for altering the input and output between the table sizing method and the table cell sizing method to render auto-sized textual mark-up language table cells with a vertical character flow property, the method comprising:  
determining that a call to a table cell sizing method from a table sizing method is a request to determine an acceptable minimum width table cell measurement for a given table cell and the request includes a minimum width input

designed to be used by a table cell element with the horizontal character flow property to render a minimum table cell width;  
determining that the given table cell content element has the vertical character flow property;  
increasing the minimum width input enough so that it is likely that all characters or objects within each paragraph in the table cell will be flowed into a separate single vertical line; and  
calling the table cell sizing method with the increased input.

Applicants respectfully submit that the Office has failed to carry the burden of establishing obviousness, because the cited art of record fails to teach or suggest “*determining that a call to a table cell sizing method from a table sizing method is a request to determine an acceptable minimum width table cell measurement for a given table cell and the request includes a minimum width input designed to be used by a table cell element with the horizontal character flow property to render a minimum table cell width; [and] determining that the given table cell content element has the vertical character flow property*”. For example, the Office directs Applicants to the following passages in CSS,

...

### 2.3 Horizontal text in vertical layout

In East Asian documents, it is often preferred to display certain Latin-based strings, such as numerals in a year, always in a horizontal layout flow regardless of the flow mode of the line of text these strings appear in, as in:

[diagram omitted]

This effect is known as “Tate naka yoko”. In order to achieve it, the Latin string should be enclosed within a SPAN element with a ‘layout-flow: horizontal’ setting in CSS, as in:

<SPAN style=“layout-flow: horizontal”>1996</SPAN>

Also, line breaking is normally disabled for such runs of text.

This can be accomplished using the CSS “white-space: nowrap” setting [CSS2].

### 2.4 Relationship with Bidirectionality

The different layout flows discussed in the preceding sections determine the text flow independently of the inherent directionality of the content characters. This means that, unless special formatting is applied to them, Hebrew and Arabic characters will be read from right to left in a horizontal layout flow and bottom to top in the ‘vertical-ideographic’ layout flow.

The `dir` attribute will affect the base direction of the element that it is applied to, but will not affect the line to line flow. For example, an

*CSS, page 7.*

...

- strict - grid type traditionally used in Chinese, as well as occasionally in Japanese (a.k.a. “Genko”) and Korean. This mode applies in the following ways to the various character sets:
  - each fullwidth character (as well as halfwidth kana) that can fit within a single grid space is rendered in the horizontal center of the grid space. The width of the grid space is determined by the ‘layout-grid-char’ setting.
  - consecutive halfwidth and cursive characters are treated as a single strip, which is then placed in the center of the smallest number of grid spaces necessary for it to fit in. They behave as if the ‘layout-grid-mode’ property were set to ‘none’ or ‘line’ around them. If a line break occurs within such a strip, the strip is treated as two separate strips whose individual placement follows the same rules as those for a single strip.
  - non-breakable objects (e.g. images) and fullwidth characters that for some reason are wider than a single grid space, are each centered within the smallest number of grid cells necessary for it to fit in.

*CSS, page 10.*

...

[diagram omitted]

Possible values:

- none - no line grid is specified.
- auto - the grid line size is determined by the largest character in the element’s font.
- <length> - size of the line grid’s unit space (a.k.a. “line pitch”).
- <percentage> - size of the line grid’s unit space relative to the logical height of the parent element.

The following markup:

```
DIV.section1 { layout-grid-line: .5in }
```

would make each line of text in a **HORIZONTALLY** (including ‘-ideographic’) laid out section of a document to be

rendered within 0.5 inch of vertical space. It is also equivalent to having a line-height of 0.5 in, as shown below:

[diagram omitted]

If the section's layout flow is **VERTICAL** (including '-ideographic'), then 0.5in is the width of each column of vertical text. This time, the 0.5in value applies to the 'width' of each cell:

[diagram omitted]

If the author preferred a specific number of lines (20 for example) to appear in an element, he would use a percentage value:

```
DIV.section1 { layout-grid-line: 5% }
```

### 3.4 'layout-grid-char'

**Value:** none | auto | <length> | <percentage>

**Initial:** none

**Applies to:** block-level elements

**Inherited:** yes

**Percentage values:** relative to element width

This property affects the dimension perpendicular to that controlled by 'layout-grid-line'. It controls the character (or "horizontal", if in horizontal layout) grid size for an element if the 'layout-grid-type' property is set to 'strict' or 'fixed'.

However, if 'layout-grid-type' is 'loose', then this property sets the size of the increment added to each fullwidth character, and, indirectly, of that added to each halfwidth character, as per the description in the specification of 'layout-grid-type'. Its effect in 'loose' grid is somewhat similar to the effect of the 'letter-spacing' property.

Note that in order for this property to have an effect, 'layout-grid-mode' must be set to 'char' or 'both'.

Possible values:

- none - no character grid.
- auto - the width of the widest character in the element's font is used as the character grid.
- <length> - width of the grid's unit space.
- <percentage> - width of the grid's unit space relative to the logical width of the parent element.

```
DIV.section1 { layout-grid-char: .5in }
```

would make each character in a horizontally laid out part of a document rendered within 0.5 inch of horizontal space:

[diagram omitted]

If the section's layout flow is vertical, then 0.5in becomes the vertical distance between consecutive characters in a column:

[diagram omitted]

If the author preferred a specific number of characters (5 for example) to appear in a line, he would set the character grid to a percentage value:

DIV.section1 { layout-grid-char: 20% }

### 3.5 'layout-grid-mode'

*CSS, pages 13-15.*

The recited passages in CSS fail to teach or suggest “*determining that a call to a table cell sizing method from a table sizing method is a request to determine an acceptable minimum width table cell measurement for a given table cell and the request includes a minimum width input designed to be used by a table cell element with the horizontal character flow property to render a minimum table cell width; [and] determining that the given table cell content element has the vertical character flow property*”.

*For example, CSS fails to teach or suggest “a table cell sizing method”, “a table sizing method”, “a call to the table cell sizing method from a table sizing method”, “determining that... a request includes a minimum width input designed to be used by a table cell element with the horizontal character flow property”, or “determining that the given table cell... has the vertical character flow property.”* CSS is a discussion about a “common formatting model” (CSS, page 1), but fails to teach or suggest anything about how the model is implemented. CSS certainly fails to teach or suggest anything about “minimum width inputs designed to be used by a table cell element with horizontal character [and] determining that the given table cell... has the vertical character flow property.”

Next, the Examiner directs Applicants to the following passage in Excel,

...

#### Aligning Items within Cells

*Alignment* refers to the positioning of characters within the cell. By default, text is left-aligned and numbers are right-aligned. To change the alignment within a cell, choose ➤ Cells, then select the Alignment tab to display the dialog box shown in Figure 5.3. The horizontal and vertical alignment of text in selected cells is controlled by two list boxes.

*Excel, page 129.*

...

#### Centering across Selections

The *Center Across Selection* option is very useful for titles, as it centers the text across the selected cells regardless of varying column widths. For example, suppose you want to center the

title *1998 Quarterly Review* over columns A through F (in row 2), as shown here:  
[diagram omitted]

### **Centering and Merging**

The old Center Across Columns button on the formatting toolbar is now called the *Merge and Center* button. This is because of a new function added to the tool, the ability to merge cells. The tool not only horizontally centers cell contents across the selected cells, but also merges the selected cells.

Merged cells act as one big cell, and do not affect the formatting of surrounding cells. They are particularly useful for labeling ranges of cells. To separate cells, first select them, and then select Format > Cells, click on the Alignment tab, and deselect the Merge Cells option. Chapter 10 discusses merged cells in more detail.

### **Controlling Vertical Alignment**

The Vertical list box controls alignment between the top and bottom of the cell. These are options:

**Top:** Positions contents at the top of the cell

**Center:** Centers contents vertically within the cell

**Bottom:** Positions contents on the bottom of the cell

**Justify:** Justifies lines vertically, from top to bottom of the cell, and automatically wraps text.

### **Reapplying the Center Across Cells Format**

Suppose you have centered a worksheet title across six cells using the Center Across Cells format. Now you want to re-center the title across just five cells. The common mistake most users make is to select the five cells, then apply the Center Across Cells format. But this won't work, because when you apply the Center Across Cells format to a range, the format is applied to each cell in the range individually. The sixth cell will retain the center-across format until you specifically remove it.

The easiest way to change the centering is to select the sixth cell and remove the center-across format (click on the Center Across Cells tool to toggle the format off). To add a cell to the center-across range, apply the Center Across Cells format to the next cell on the right side of the range. To remove a cell from the center-across range, remove the Center Across Cells format from the last cell on the right side of the range.

To demonstrate the effect of vertical alignment, try this exercise:

1. Enter some text in cell B2.
2. Increase the height of row 2, say to 50 or so.
3. Change the vertical alignment in cell B2 to Top.

Notice that the vertical alignment in a cell is not apparent unless the row height is increased.

**Changing Text Appearance with Text Control**

The Text Control section of the Alignment tab changes several aspects of the way text appears in a cell or group of cells, and on the worksheet. These are the text control options:

**Wrap Text:** The Wrap Text option breaks a long line of text into multiple lines to fit within the cell. Excel breaks the lines to fit column width, but you can insert specific line breaks with a “soft” return, Alt+↵. Row height automatically increases to fit multiple lines of text.

[diagram omitted]

**Shrink to Fit:** The Shrink To Fit option changes the font size instead of the row height, shrinking the text to fit in the cell. This is a handy feature when text is just a little too bit for the selected cell, as shown above, but if you try to shrink too much text without changing the cell size, the text can be too small to read without zooming.

**Merged Cells:** The Merged Cells option is an extremely useful new feature. It allows you to merge several cells together to act as one, without changing row heights and column widths. By merging cells, you can place addresses, labels, or explanatory text on worksheets, and maintain the formatting integrity of accompanying data. This is especially useful when you are creating documents such as invoice templates. In the image shown below, cells B3:D6 have been merged to create space for the company address, but the widths of the adjoining columns of data are intact.

...

*Excel, pages 131-133.*

The recited passages in Excel fail to teach or suggest “*determining that a call to a table cell sizing method from a table sizing method is a request to determine an acceptable minimum width table cell measurement for a given table cell and the request includes a minimum width input designed to be used by a table cell element with the horizontal character flow property to render a minimum table cell width; [and] determining that the given table cell content element has the vertical character flow property*”.

*For example, Excel fails to teach or suggest “a table cell sizing method”, “a table sizing method, a call made there between”, “a minimum width input designed to be used by a table cell element with the horizontal character flow property”, or “the given table cell... has*

***the vertical character flow property.” Excel certainly fails to teach anything about “a minimum width input designed to be used by a table cell element with the horizontal character flow property” or that the “given table cell... has the vertical... property.”***

To establish a prima facie case of obviousness, the Office must direct applicants to references that teach or suggest all of a claim’s limitations. Applicants respectfully submit that the Office has failed to carry the burden of establishing obviousness, because the cited art of record fails to teach or suggest “*determining that a call to a table cell sizing method from a table sizing method is a request to determine an acceptable minimum width table cell measurement for a given table cell and the request includes a minimum width input designed to be used by a table cell element with the horizontal character flow property to render a minimum table cell width; [and] determining that the given table cell content element has the vertical character flow property*”.

For at least this reason amended claim 1 is in condition for allowance. Such action is respectfully requested.

#### ***Claims 2-5***

Claims 2-5 depend from claim 1. Since they depend from claim 1, they should be allowed for at least the reasons stated for claim 1. In view of the foregoing discussion of claim 1, the merits of the separate patentability of dependent claims 2-5 are not belabored at this time. Claims 2-5 should be allowable. Such action is respectfully requested.

#### ***Independent Claims 6, 11, 21, 22, 25, 28, and 29***

Applicants respectfully submit that for reasons similar to those stated above, the CSS-Excel combination fails to teach or suggest the following features:

*Claim 6* - “determining that a call to a table cell sizing method from a table sizing method is a request to determine an acceptable maximum width table cell measurement for a given table cell and the request includes a maximum width input designed to be used by a table cell element with the horizontal character flow property to render a maximum table cell width; determining that the given table cell content element has the vertical character flow property”.

*Claim 11* - “estimating a logical width for input to a table cell sizing method designed to auto-size a table cell with the horizontal character flow property; and

calling the table cell sizing method with the estimated logical width input in order to auto-size the table cell with the vertical character flow property.”

*Claim 21* - “a second program module designed to accept requests to auto-size a textual mark-up language element with the horizontal character flow property at a requested horizontal width;

a third program module designed to alter the request made to the second program module by the first program module so that while the second program module continues to operate as it would for a textual mark-up language element with the horizontal character flow property, the second program module is actually auto-sizing a textual mark-up element with the vertical character flow property”.

*Claim 22* - “wherein the first module is designed to request auto-sizing of table cell elements with the horizontal character flow property and the third module alters inputs of the first module’s requests to the second module when the request is to auto-size a table cell element with the vertical character flow property.”

*Claim 25* - “wherein the second module is designed to receive requests to auto-size table cell elements with a horizontal character flow property and the third module alters an input of a request from the first module to the second module when the request is to auto-size a table cell element with the vertical character flow property.”

*Claim 28* - “a second object designed to accept requests to auto-size a textual mark-up language element with the horizontal character flow property at a defined horizontal width; and

a third object designed to alter the request made to the second object by the first object so that while the second object continues to operate as it would for a textual mark-up language element with the horizontal character flow property, the second object is actually auto-sizing a textual mark-up element with the vertical character flow property.”

*Claims 29* - “second object returns the resulting height and width measurements of the table cell element to the third object... the third object forwards the returned height measurement to the first object as the measurement for the maximum table cell width.

Since the CSS-Excel combination fails to teach or suggest these features of independent claims 6, 11, 21, 22, 25, 28, and 29, they should be allowable. Such action is respectfully requested.

***Dependent Claims 7-10, 12-17, 23-24, 26-27, and 30-32***

Claims 7-10, 12-17, 23-24, 26-27, and 30-32 depend from the above allowable independent claims. since claims 7-10, 12-17, 23-24, 26-27, and 30-32 depend from the above allowable independent claims, they should be allowed for at least the above reasons. Such action is respectfully requested.

***Claim 18***

Applicants respectfully assert that the Office has failed to carry the burden of establishing a prima facie case of obviousness, because the art of record fails to teach or suggest *"multiplying the obtained actual logical height by the obtained actual logical width in order to determine the area of the maximum logical height rectangle"*.

Specifically, claim 18 recites,

18. (original) In a computer system with a textual mark-up language engine, a parent textual mark-up language element with a horizontal character flow property, a child textual mark-up language element with a vertical character flow property, a method for determining a proposed logical width dimension, the method being given a proposed physical width, a desired layout area, and the child element, the method comprising:

determining an area of a minimum logical height rectangle for the child element by flowing each paragraph or sentence in the child element into a single line in the character flow direction; and

obtaining an actual logical width of a minimum logical height

rectangle by measuring a length of a longest line in the child element in the character flow direction;

obtaining an actual logical height of a minimum logical height

rectangle by finding and summing the measurements of each character or object in each said line that occupies the most space in the direction perpendicular to the character flow direction and adding a minimal space between lines to improve readability; and

multiplying the obtained actual logical height by the obtained

actual logical width;

determining an area of a maximum logical height rectangle for the child element by;

setting a maximum logical width for the maximum  
logical height  
rectangle;  
flowing the child element into the maximum logical  
height  
rectangle in a line in the character flow direction so  
long as no next character or object in the content element being  
flowed into the line would cause the length of the line to  
exceed the set maximum logical width;  
starting a new line next to and parallel to the previous  
line and  
flowing the content element into that line each time the  
next character or object in the child element would cause the  
line then being flowed into the rectangular area to exceed the  
set maximum logical width;  
obtaining an actual logical width of the maximum  
logical  
height rectangle by finding and measuring a length of  
the longest line in the character flow direction;  
obtaining an actual logical height of the minimum  
logical height  
rectangle by finding and summing the measurements of  
each character or object in each said line that occupies the most  
space in the direction perpendicular to the character flow  
direction and adding a minimal space between lines to improve  
readability;  
multiplying the obtained actual logical height by  
the obtained actual logical width in order to determine  
the area of the maximum logical height rectangle;  
determining a proposed logical width by summing the  
areas of the determined maximum logical height rectangle and  
the determined minimum logical height rectangle and dividing  
the sum by approximately twice the proposed physical width.

Applicants respectfully submit that the Office has failed to carry the burden of  
establishing obviousness, because the cited art of record fails to teach or suggest “*multiplying the  
obtained actual logical height by the obtained actual logical width in order to determine the  
area of the maximum logical height rectangle*”. For example, the Office directs Applicants to  
the following passages in CSS,

...  
[diagram omitted]  
Possible values:

- none - no line grid is specified.

- auto - the grid line size is determined by the largest character in the element's font.
- <length> - size of the line grid's unit space (a.k.a. "line pitch").
- <percentage> - size of the line grid's unit space relative to the logical height of the parent element.

The following markup:

```
DIV.section1 { layout-grid-line: .5in }
```

would make each line of text in a **HORIZONTALLY** (including '-ideographic') laid out section of a document to be rendered within 0.5 inch of vertical space. It is also equivalent to having a line-height of 0.5 in, as shown below:

[diagram omitted]

If the section's layout flow is **VERTICAL** (including '-ideographic'), then 0.5in is the width of each column of vertical text. This time, the 0.5in value applies to the 'width' of each cell:

[diagram omitted]

If the author preferred a specific number of lines (20 for example) to appear in an element, he would use a percentage value:

```
DIV.section1 { layout-grid-line: 5% }
```

### 3.4 'layout-grid-char'

**Value:** none | auto | <length> | <percentage>

**Initial:** none

**Applies to:** block-level elements

**Inherited:** yes

**Percentage values:** relative to element width

This property affects the dimension perpendicular to that controlled by 'layout-grid-line'. It controls the character (or "horizontal", if in horizontal layout) grid size for an element if the 'layout-grid-type' property is set to 'strict' or 'fixed'. However, if 'layout-grid-type' is 'loose', then this property sets the size of the increment added to each fullwidth character, and, indirectly, of that added to each halfwidth character, as per the description in the specification of 'layout-grid-type'. Its effect in 'loose' grid is somewhat similar to the effect of the 'letter-spacing' property.

Note that in order for this property to have an effect, 'layout-grid-mode' must be set to 'char' or 'both'.

Possible values:

- none - no character grid.
- auto - the width of the widest character in the element's font is used as the character grid.
- <length> - width of the grid's unit space.

- <percentage> - width of the grid's unit space relative to the logical width of the parent element.

DIV.section1 { layout-grid-char: .5in }

would make each character in a horizontally laid out part of a document rendered within 0.5 inch of horizontal space:

[diagram omitted]

If the section's layout flow is vertical, then 0.5in becomes the vertical distance between consecutive characters in a column:

[diagram omitted]

If the author preferred a specific number of characters (5 for example) to appear in a line, he would set the character grid to a percentage value:

DIV.section1 { layout-grid-char: 20% }

### 3.5 'layout-grid-mode'

CSS, pages 13-15.

The recited passages in CSS fail to teach or suggest "*multiplying the obtained actual logical height by the obtained actual logical width in order to determine the area of the maximum logical height rectangle*".

Next, the Examiner directs Applicants to the following passage in Excel,

...

#### Aligning Items within Cells

*Alignment* refers to the positioning of characters within the cell.

By default, text is left-aligned and numbers are right-aligned.

To change the alignment within a cell, choose ➤ Cells, then select the Alignment tab to display the dialog box shown in Figure 5.3. The horizontal and vertical alignment of text in selected cells is controlled by two list boxes.

*Excel, page 129.*

...

#### Centering across Selections

The *Center Across Selection* option is very useful for titles, as it centers the text across the selected cells regardless of varying column widths. For example, suppose you want to center the title *1998 Quarterly Review* over columns A through F (in row 2), as shown here:

[diagram omitted]

#### Centering and Merging

The old Center Across Columns button on the formatting toolbar is now called the *Merge and Center* button. This is because of a new function added to the tool, the ability to merge cells. The tools not only horizontally centers cell contents across the selected cells, but also merges the selected cells.

Merged cells act as one big cell, and do not affect the formatting of surrounding cells. They are particularly useful for labeling ranges of cells. To separate cells, first select them, and then select Format > Cells, click on the Alignment tab, and deselect the Merge Cells option. Chapter 10 discusses merged cells in more detail.

### **Controlling Vertical Alignment**

The Vertical list box controls alignment between the top and bottom of the cell. These are options:

**Top:** Positions contents at the top of the cell

**Center:** Centers contents vertically within the cell

**Bottom:** Positions contents on the bottom of the cell

**Justify:** Justifies lines vertically, from top to bottom of the cell, and automatically wraps text.

### **Reapplying the Center Across Cells Format**

Suppose you have centered a worksheet title across six cells using the Center Across Cells format. Now you want to re-center the title across just five cells. The common mistake most users make is to select the five cells, then apply the Center Across Cells format. But this won't work, because when you apply the Center Across Cells format to a range, the format is applied to each cell in the range individually. The sixth cell will retain the center-across format until you specifically remove it.

The easiest way to change the centering is to select the sixth cell and remove the center-across format (click on the Center Across Cells tool to toggle the format off). To add a cell to the center-across range, apply the Center Across Cells format to the next cell on the right side of the range. To remove a cell from the center-across range, remove the Center Across Cells format from the last cell on the right side of the range.

To demonstrate the effect of vertical alignment, try this exercise:

1. Enter some text in cell B2.
2. Increase the height of row 2, say to 50 or so.
3. Change the vertical alignment in cell B2 to Top.

Notice that the vertical alignment in a cell is not apparent unless the row height is increased.

### **Changing Text Appearance with Text Control**

The Text Control section of the Alignment tab changes several aspects of the way text appears in a cell or group of cells, and on the worksheet. These are the text control options:

**Wrap Text:** The Wrap Text option breaks a long line of text into multiple lines to fit within the cell. Excel breaks the lines to fit column width, but you can insert specific line breaks with a "soft" return, Alt+↵. Row

height automatically increases to fit multiple lines of text.

[diagram omitted]

**Shrink to Fit:** The Shrink To Fit option changes the font size instead of the row height, shrinking the text to fit in the cell. This is a handy feature when text is just a little too bit for the selected cell, as shown above, but if you try to shrink too much text without changing the cell size, the text can be too small to read without zooming.

**Merged Cells:** The Merged Cells option is an extremely useful new feature. It allows you to merge several cells together to act as one, without changing row heights and column widths. By merging cells, you can place addresses, labels, or explanatory text on worksheets, and maintain the formatting integrity of accompanying data. This is especially useful when you are creating documents such as invoice templates. In the image shown below, cells B3:D6 have been merged to create space for the company address, but the widths of the adjoining columns of data are intact.

...

*Excel, pages 131-133.*

The recited passages in Excel fail to teach or suggest “*multiplying the obtained actual logical height by the obtained actual logical width in order to determine the area of the maximum logical height rectangle*”.

To establish a prima facie case of obviousness, the Office must direct applicants to references that teach or suggest all of a claim’s limitations. Applicants respectfully submit that the Office has failed to carry the burden of establishing obviousness, because the cited art of record fails to teach or suggest “*multiplying the obtained actual logical height by the obtained actual logical width in order to determine the area of the maximum logical height rectangle*”.

For at least this reason amended claim 18 is in condition for allowance. Such action is respectfully requested.

#### ***Claims 19-20***

Claims 19-20 depend from claim 18. Since they depend from claim 18, they should be allowed for at least the reasons stated for claim 18. In view of the foregoing discussion of claim 18, the merits of the separate patentability of dependent claims 19-20 are not belabored at this time. Claims 19-20 should be allowable. Such action is respectfully requested.

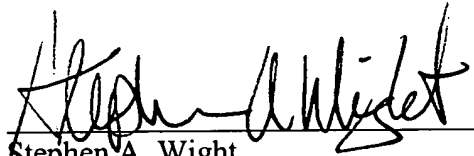
**Conclusion**

The claims in their present form should now be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

By

  
Stephen A. Wight  
Registration No. 37,759

One World Trade Center, Suite 1600  
121 S.W. Salmon Street  
Portland, Oregon 97204  
Telephone: (503) 226-7391  
Facsimile: (503) 228-9446